Haylee Ham

Proposing an Observational Study

As humans develop skills, do they continue to face greater challenges and become more frustrated with their work or are they better equipped to tackle those challenges and therefore get more enjoyment out of their work? Furthermore, when a human feels frustrated with their method of solving a problem, will they alter their method or do they feel too invested to make a change? I will examine these question by looking at the emotional responses of programmers and the change in usage of languages (the methods programmers use to solve problems) in light of the amount of frustration those languages elicit.

To do so, I will research the emotions present in commit messages from users in the GitHub Archive dataset over languages and over time. I would like to see if a higher prevalence of anger results in lower usage of that language and which emotion (anger or joy) becomes more prevalent the longer a user is using GitHub. I will parse through the commit messages in order to flag emotional responses of anger and joy and then use the metadata to collect time of commit and language of data being committed.

Since all commit messages that are not part of a private repository are public, there are ample commit messages in each major language subset (at least 100,000). It may be that in shared professional repositories, emotional responses are dampened due to this professional environment, however, due to the vast size of the dataset, significance can still be found in the remaining more casual commit messages.

Using an R sentiments analysis package, I will categorize commit messages into groups such as neutral, joyful, and angry. This analysis will use lists of words that correspond to major emotions and then categorize the messages. Due to the short nature of commit messages, each message usually contains only one sentiment, which will be helpful in this analysis.

My study will use the measuring method in order to quantify the amount of anger or joy in commit messages across languages and across time. Since GitHub has millions of datapoints, each with a snapshot of the programmer's mental state upon committing, I will not need to use any observational methods other than measuring.

Forecasting, for instance, is inappropriate here as I want to study these effects in already collected data over time rather than attempting to predict what one language's popularity will be tomorrow or how one programmer will feel next week.

In the presence of this dataset, it is also unnecessary for me to use the methods of a natural experiment or matching. GitHub users are not randomly assigned to program in certain languages and therefore I cannot consider the language used to be a treatment inducing any particular outcome. In the case of matching, as all GitHub users are using one language or another it is also not possible to match a user using one language with a user who has received no treatment in order to measure the effect. All of the data points in the GitHub dataset have

experienced a treatment as all data points are instances of users coding in a language and then committing that code to a repository.

This dataset is undeniably big. The GitHub Archive data contains 2.8 million open repositories, 145 million unique commits, and metadata for all observations. Due to this bigness, I will be able make estimates for subgroups such as commit messages for each specific language. I will also be able to distinguish small differences in emotional responses over time and languages. Since this dataset is a collection of all commits over the lifetime of GitHub, it can also be said to be always-on. Because of this, the data will contain unexpected events (such as bothersome bugs or fixes in languages that caused a large emotional response). Perhaps the most interesting characteristic of big data in this study is that it is non-reactive. Presumably, GitHub users mostly take some degree of privacy for granted. Their code and commit messages are open source but, given the sheer volume and tedium of commit messages, users feel very little social desirability bias. This results in commit messages that are generally unfiltered and contain emotional outbursts.

One the other hand, I will need to address some difficulties that this big dataset poses to my research question. First, the GitHub Archive is incomplete. There is no demographic data in the dataset. Lacking information on gender, nationality, age, and race will not largely affect my research question since I am interested in measuring the emotions felt by users of GitHub over time and by language in the aggregate. Additionally, GitHub users always have the option to make a repository private, in which case data about commits to those repositories are not included in the dataset. It could be argued that GitHub users most freely express their emotions in commits to private repositories since they are sure no one will ever see them, however since the ability to make a repository private is a paid feature, it is probable that the majority of users keep their repositories public.

Second, the data may be non-representative. Reportedly, GitHub users are about 70% male and 60% of users are under 35, therefore any conclusions about emotions toward programming and different languages over time cannot be applied to the average programmer as much as to the average GitHub user. GitHub is very widely used by open-source language and project developers, hobbyists, students, and professionals. However it was launched only in 2008 and is largely skewed toward the young and internet savvy. This may result in findings that show more anger towards languages that are not widely used by younger programmers. However, this should have no effect on the analysis of emotional response in commit messages over time.

Third, this dataset is somewhat dirty. I will be looking predominantly at commit messages, which supposedly reflect the work done by the programmer since the last commit. However, it is possible to automate commits and therefore emotional responses will not be present in these automated commit messages. In order to compensate for this, I will attempt to remove as many automated messages as possible. There are several popular scripts that automate commits for a GitHub user and each of these scripts has a default commit message. For example,

one popular script sets the commit message to be “gf”. With this knowledge, I can remove as many automated commit messages from the dataset as possible.

Fourth, the GitHub data is sensitive. Included in the dataset is the username and email address of the author of the code. Before working with the dataset, I would remove unnecessary identifying information (such as email address) and set the usernames to be a string of integers so that a single user could be tracked across time but not identified. All information will be dealt with in the aggregate and there will be no identifying or individual-level information in my research paper.

Finally, I will address the feasibility of using this research question for my MA thesis. The GitHub Archive dataset is already publicly available through Google BigQuery. The dataset includes all information for commit messages, languages used, and code pushed for each public repository in GitHub. Due to the somewhat sensitive nature of this data, I do plan to anonymize the usernames as mentioned above and I will not be including any individual-level code or information in my research paper lest it be identifiable.